IEEE 754 Decimal Floating-Point —— in Binary

John Crawford
Intel Fellow
Intel Corporation

April 26, 2006

Thanks to:
Ping Tak Peter Tang, Ricardo Morin,
Mahesh Bhat (Intel)
Mike Cowlishaw (IBM)

Agenda

- **Why Decimal FP?**
- **Natural Advantages, Intel's View**
- **Tale of Two Formats**
- **Summary**

Why not use binary FP?

- binary fractions *cannot* exactly represent all decimal fractions

- 1.2 x 1.2 → 1.44 ?

  – 1.2 in a 32-bit binary float is actually:
  1.200000047683715820312 5

  – and this squared is:
  1.440000057220458984375

Source: Mike Cowlishaw
http://www2.hursley.ibm.com/decimal/decArith16-charts.pdf

A financial example…

- 5% sales tax on a $ 0.70 telephone call, rounded to the nearest cent
- 1.05 x 0.70 using binary double type is

  0.73499999999999986677323704498121514916419982910156 25

  (should have been 0.735)
- rounds to $ 0.73, instead of $ 0.74

Source: Mike Cowlishaw
http://www2.hursley.ibm.com/decimal/decArith16-charts.pdf

Hence…

- Binary floating-point cannot be used for commercial applications
  - cannot match values computed by hand
  - cannot meet legal and financial requirements, which are based on 2,500+ years of decimal arithmetic
- So applications use decimal software floating-point packages…

Source: Mike Cowlishaw
http://www2.hursley.ibm.com/decimal/decArith16-charts.pdf

Agenda

- Why Decimal FP?
- **Natural Advantages, Intel's View**
- Tale of Two Formats
- Summary

Decimal FP Pros/Cons

- Advantage: Match Pencil & Paper Calculations
  - Decimal Fractions represented exactly
  - Wide precisions (16 and 34 digits) support exact calculation (no rounding) for modest formulas
- Disadvantage: Worst-Case Rounding error
  - Coarser rounding boundary, “Wobble”
- Disadvantage: Efficiency
  - Decimal FP logic is larger, consumes more power, and longer latency than Binary FP logic
  - Not quantified yet

Intel's View of IEEE 754r Decimal FP

- Decimal Types are Important
  - Most monetary data in app code is naturally decimal
  - Standardized computation is important – same results
- Applications Experience with Software Decimal FP
  - Decimal cycle % is modest (<5%) in Monetary Apps
- IEEE 754r must enable efficient software library
  - Satisfies the broad market, for the foreseeable future
  - Needed to start adoption of the standard
- Formats must be software friendly
  - Efficient unpacking of components into binary
  - Exploit existing binary hardware
  - Efficient packing of components back to compact form

Agenda

- Why Decimal FP?
- Natural Advantages, Intel's View
- **Tale of Two Formats**
- Summary

Decimal FP Encoding – DPD

Encoding numeric values

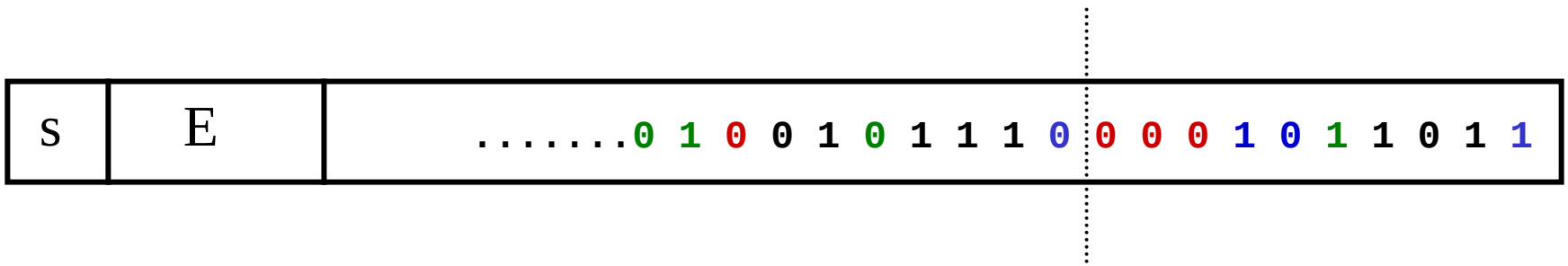
$$(-1)^s \, 10^{E\text{-bias}} \, c$$

For example 8280.95 = $10^{-2} \times$ 8 2 8 0 9 5

Conceptually BCD 1000 0010 1000 0000 1001 0101

Packed into DPD 01 0 01 0 111 0 000 10 1 101 1

indicator indicator



Decimal FP Encoding – BID

Encoding numeric values

$$(-1)^s \, 10^{E\text{-bias}} \, c$$

For example 8280.95 = $10^{-2} \times$ 8 2 8 0 9 5

11001010001010111111

s	E	.....1 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 1

Software Performance of BID

1.5GHz Itanium2	BID	DPD
TOTAL TIME (sec)	0.40	1.01
Decimal Time (sec)	0.15	0.65
latencies (avg. in cycles)		
add	12	88
mul	12	95
rescale	40	101

Factors contributing to BID performance gap

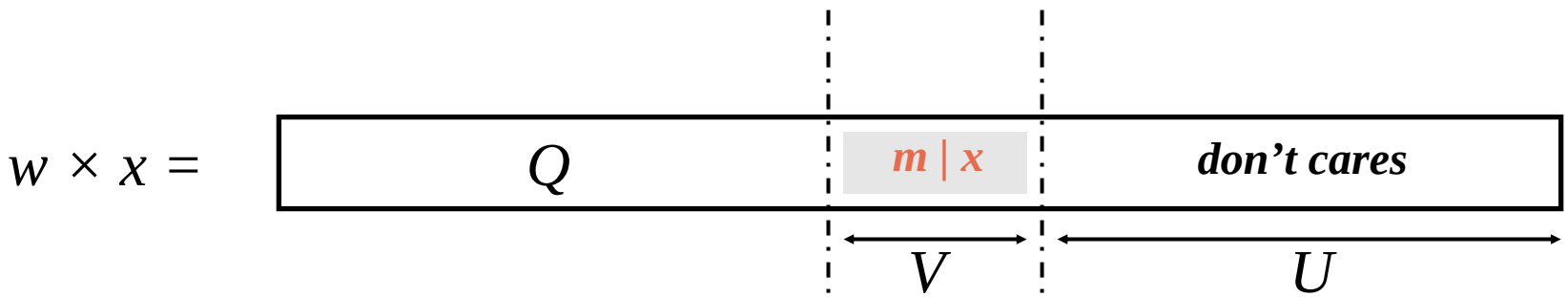
- Binary algorithms – use existing binary hardware
- Small pack/unpack overhead for BID <-> binary components

Decimal Rounding BID

Let $0 < x < 2^U$ be a dividend and $0 < m < 2^V$ be a divisor.

Let $x/m = Q + (r/m)$, we wish to compute Q and $r == 0$

Let $S = U + V$ and $w = \text{ceil}(2^S / m) = 2^S / m + a/m, \quad 0 < a < m$



Evolution Key to Decimal

- BID is compatible with binary hardware
- Incremental hardware roadmap possible

Microcode + ROM

Existing multiplier, multiple pass (non-pipelined)

Existing multiplier with separate non-pipelined rounder

Existing multiplier with pipelined rounder

Separate unit

Summary

- Decimal FP is important for Monetary Computations
  - Data is naturally decimal
  - Legally Required
  - Modest, but not negligible, profile in decimal operations
- Efficient Software Library
  - Satisfies the Broad Market, for the foreseeable future
  - Needed to foster adoption of the standard
- Binary-Integer Decimal (BID) Format Enables
  - An efficient software library
  - Efficient Rounding with proper algorithms
  - Hardware sharing with binary FP
  - Phased hardware implementation

Disclaimer

- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit http://www.intel.com/performance/resources/limits.htm.